A Novel Performance Model of J2EE Web Applications Considering Application Server Settings

Gábor Imre and Hassan Charaf

Managing business processes, the improper performance of a web application can cause serious financial loss to a company. The performance-related requirements of an Internet application are often recorded in a Service Level Agreement (SLA). SLAs can specify an upper limit for the average response time, a lower limit for availability, while the application guarantees a certain throughput level.

These performance metrics depend on several factors, such as hardware, software, network, and client workload. This paper focuses on the settings of the application server software that serves the HTTP requests of the browsers. More precisely, the performance of a test web application is measured under different client load with different values of two parameters of the application server. These tuning parameters are the maximum size of the thread pool, and the maximum size of the HTTP connection queue. In the application server, accepted HTTP connections are placed into a connection queue. The size of the connection queue is limited by an adjustable parameter of the given application server. When this limit is reached, it refuses to serve the request. The threads in the thread pool take connections from the queue and serve the requests. The server can decide to create more threads (i. e. increase the size of the thread pool), but cannot exceed a certain configurable maximum. When the maximum thread pool size is reached, however, the requests are not dropped, as long as they find free space in the connection queue.

Setting up a performance model that is capable of creating a quantitative relationship between the performance factors and the performance is a key issue to meet the performance requirements of SLAs. Using queueing networks is a popular method of performance modelling. In [1], a queueing model for multi-tier internet applications is presented, that faithfully captures concurrency limits at the tiers. The maximum size of the connection queue, as presented earlier, can be considered as a concurrency limit of the web tier in this model, but it cannot handle the maximum size of the thread pool. A powerful combination of the queueing network and the Petri net formalism is presented in [2]. Using queueing Petri nets, the authors successfully model the performance of a web application, considering the maximum size of thread pools. Their model, however, does not take the maximum size of the connection queue into account.

Our paper shows that the limits configured both for the connection queue and the thread pool have a considerable effect on the performance and presents a queueing network based performance model that considers both of them.

Keywords: *performance evaluation, Web technologies, queueing networks*

References

[1] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer and A. Tantawi. An Analytical Model for Multi-Tier Internet Services and Its Applications, *ACM SIGMETRICS Performance Evaluation Review*, 33(1), 2005, pp. 291-302.

[2] S. Kounev, and A. Buchmann. Performance Modelling of Distributed E-Business Applications Using Queueing Petri Nets, in *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software*, Austin, Texas, 2003, pp. 143-155.